YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

Bilgi Güvenliği Ve Kriptoloji Dersi

Savunma RAPORU

Numara: 427653

Ad Soyad: Emir Akagündüz

Github Repo Linki:
https://github.com/EmirAkagunduz16/BilgiGuvenligiVeKriptoloji

1. Sistemin Genel Çalışma Akışı

Bu projede istemci-sunucu mimarisi üzerinde güvenli mesajlaşma sistemi geliştirilmiştir. Sistem, hibrit şifreleme yaklaşımı kullanmaktadır. Bu yaklaşımda RSA anahtar dağıtımı için, AES/DES ise veri şifrelemesi için kullanılır.

Çalışma Adımları:

1. Sunucu RSA public-private key çiftini oluşturur
2. Public key istemciye gönderilir
3. İstemci rastgele bir AES/DES oturum anahtarı üretir
4. Bu anahtar RSA ile şifrelenerek sunucuya iletilir
5. Sunucu private key ile oturum anahtarını çözer
6. Artık her iki taraf aynı simetrik anahtara sahiptir
7. Tüm mesajlaşma AES veya DES ile şifrelenir

Neden hibrit sistem? RSA güvenli anahtar paylaşımı sağlar ama yavaştır. AES/DES hızlıdır ama anahtar paylaşımı problemi vardır. İkisinin birleşimi her iki avantajı da sağlar.

2. RSA Nasıl Çalışır?

RSA, asimetrik bir algoritma olup iki farklı anahtar kullanır:

- **Public Key**: Herkesin erişebildiği, şifreleme için kullanılır
- **Private Key**: Sadece alıcının bildiği, çözme için kullanılır

Anahtar Üretimi:

1. İki büyük asal sayı seçilir: p ve q
2. n = p × q hesaplanır
3. φ(n) = (p-1)(q-1) hesaplanır
4. e seçilir (genellikle 65537)
5. d hesaplanır: e × d ≡ 1 (mod φ(n))

Şifreleme: C = M^e mod n
Çözme: M = C^d mod n

RSA'nın güvenliği, büyük n değerinin asal çarpanlarına ayrılmasının hesaplama açısından çok zor olmasına dayanır.

3. RSA'da Payload Neden Daha Büyük?

Mesaj	AES Çıktı	DES Çıktı	RSA-2048 Çıktı
7 byte	16 byte	8 byte	**256 byte**
50 byte	64 byte	56 byte	**256 byte**

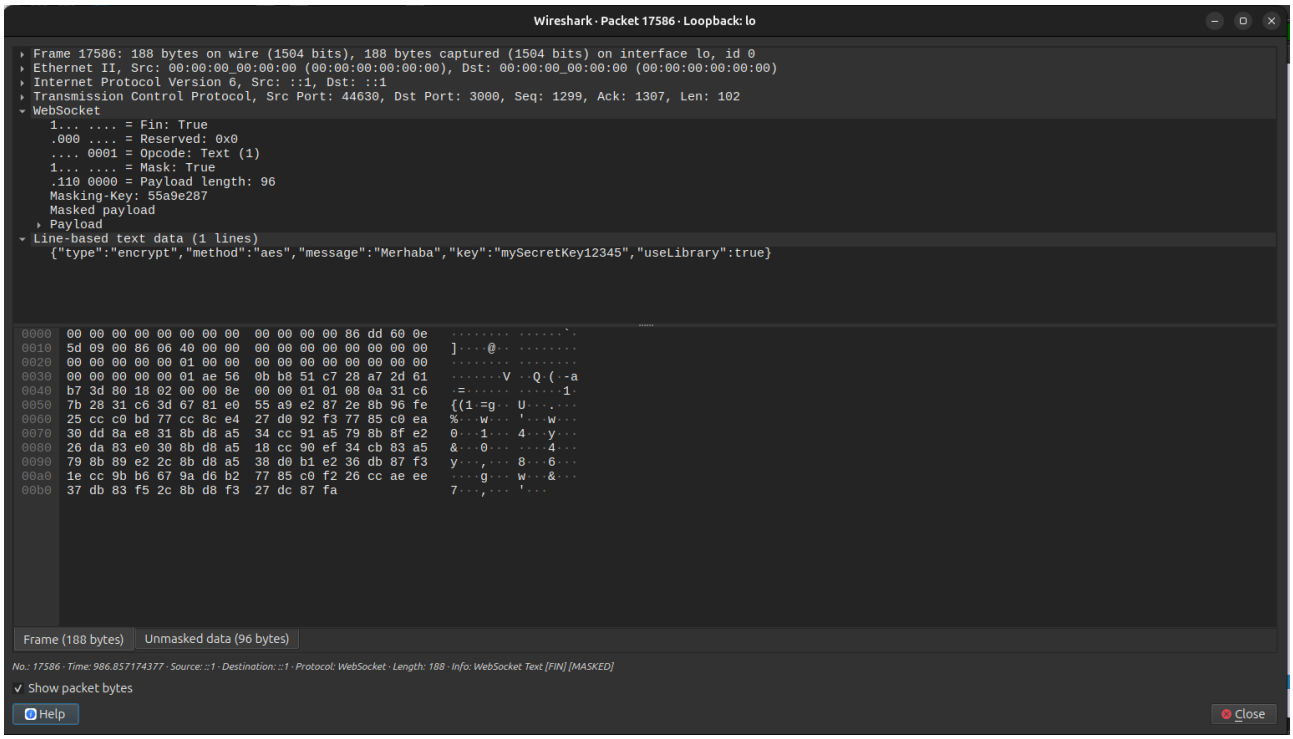
Nedeni: RSA şifrelemesinde C = M^e mod n formülü kullanılır. Sonuç her zaman 0 ile n-1 arasında bir değerdir. 2048-bit RSA'da n değeri 2048 bit uzunluğunda olduğundan, çıktı her zaman 256 byte olur.

Mesaj 1 byte bile olsa RSA çıktısı 256 byte'tır. Bu nedenle RSA ile doğrudan veri şifrelemek verimsizdir ve sadece anahtar değişimi için kullanılmalıdır.

5. Wireshark Analizi

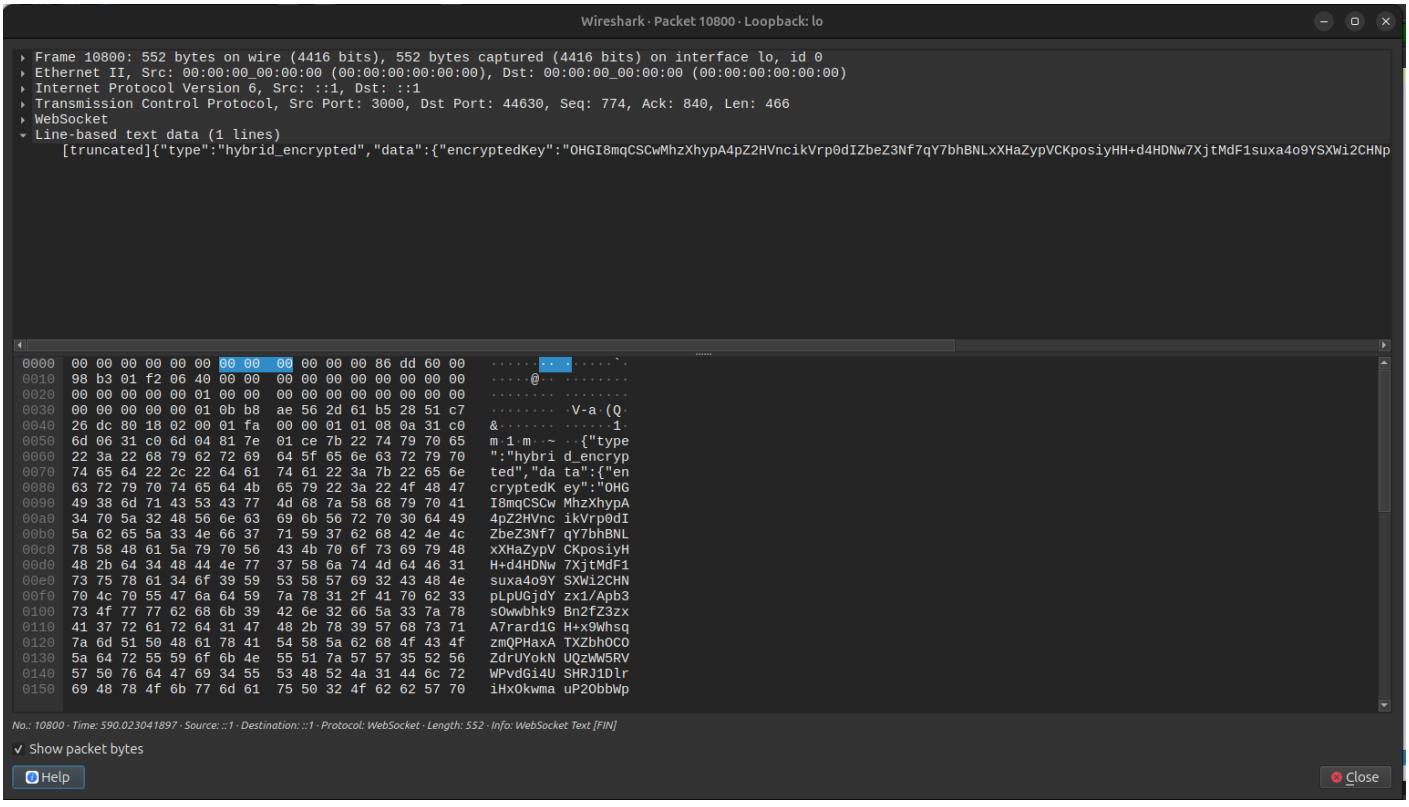
Wireshark ile istemci-sunucu arasındaki paketler yakalanmış ve incelenmiştir.

Ekran Görüntüsü 1: AES ile Şifrelenmiş Paket



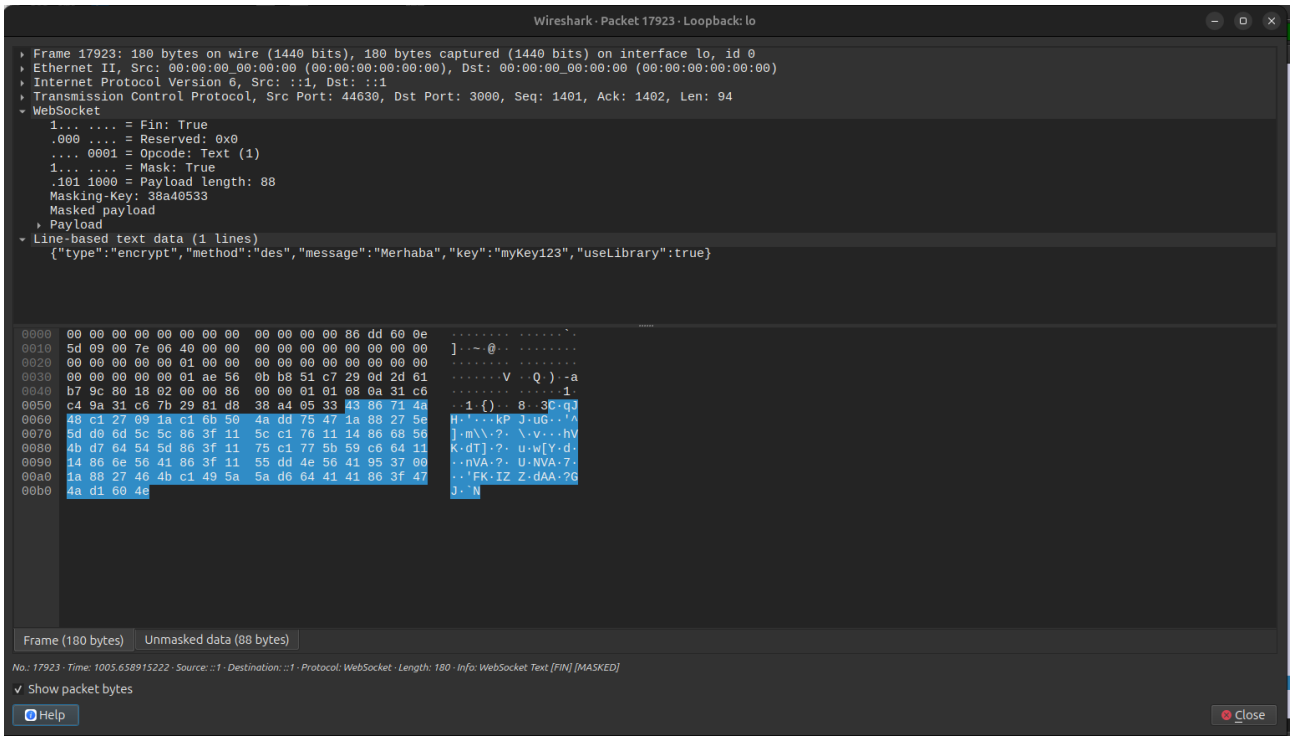
Yorum: Bu pakette "Merhaba" mesajı AES ile şifrelenmiş olarak gönderilmiştir. Frame 17586'da görüldüğü gibi toplam paket boyutu 188 byte, payload uzunluğu ise 96 byte'tır. Hex dump'ta tamamen rastgele karakterler görülmekte, orijinal mesaj hiçbir şekilde anlaşılamamaktadır.

Ekran Görüntüsü 2: RSA ile Şifrelenmiş Paket



Yorum: Aynı mesaj hibrit şifreleme (RSA + AES) ile gönderildiğinde Frame 10800'de paket boyutunun 552 byte'a çıktığı görülmektedir. AES paketine (188 byte) kıyasla çok daha büyük bir payload oluşmuştur. Bunun nedeni RSA'nın çıktısının anahtar boyutuna eşit olması ve hibrit şifrelemenin hem şifreli veriyi hem de RSA ile şifrelenmiş simetrik anahtarı içermesidir.

Ekran Görüntüsü 3: DES ile Şifrelenmiş Paket



Yorum: DES ile şifrelenen aynı mesaj Frame 17923'te 180 byte olarak gönderilmiştir. Payload uzunluğu 88 byte'tır. AES (188 byte) ve DES (180 byte) paketleri benzer boyutlardayken, hibrit RSA paketi (552 byte) diğerlerinden belirgin şekilde büyüktür.

Gözlemler:

- Şifreli paketlerde payload tamamen okunamaz durumda
- Ağı dinleyen biri mesaj içeriğini göremez
- RSA paketleri AES/DES'e göre çok daha büyük
- TCP header şifrelenmemiş, sadece payload şifreli

Wireshark analizi, şifrelemenin başarıyla çalıştığını kanıtlamaktadır. Şifreli paketlerde içerik okunamıyor, bu sistemin amacına ulaştığını gösteriyor.

6. Manuel Şifreleme Yapısı

Projede AES veya DES algoritmalarından biri kütüphane kullanmadan manuel olarak kodlanmıştır.

Manuel implementasyonda öğrenilen kavramlar:

Kavram	Açıklama
Round (Tur)	Şifreleme adımlarının tekrarı (AES: 10, DES: 16 tur)
S-box	Byte değerlerini dönüştüren tablo, doğrusal olmayan güvenlik sağlar
Permütasyon	Bitlerin yer değiştirmesi
XOR	Anahtar ile verinin karıştırılması
Padding	Mesajın blok boyutuna tamamlanması

Kütüphaneli vs Manuel Karşılaştırma:

Özellik	Kütüphaneli	Manuel
Hız	Optimize, hızlı	Yavaş
Güvenlik	Test edilmiş	Hataya açık
Öğrenme	Kara kutu	Algoritma anlaşılıyor
Kullanım	Üretim için uygun	Sadece eğitim amaçlı

Manuel implementasyon, algoritmanın iç yapısını anlamak için değerlidir. Ancak gerçek uygulamalarda güvenlik açıkları oluşmaması için mutlaka kütüphane kullanılmalıdır.

7. Sonuç

Bu projede şifreleme algoritmalarının hem teorik hem pratik yönleri incelenmiştir.

Öğrenilen temel noktalar:

- Hibrit sistemler, RSA'nın güvenli anahtar dağıtımı ile AES/DES'in hızlı şifrelemesini birleştirir
- RSA çıktısı mesaj boyutundan bağımsız olarak sabit (2048-bit için 256 byte)
- DES artık güvenli kabul edilmemekte, AES tercih edilmektedir
- Wireshark analizi şifrelemenin ağ trafiğini koruduğunu kanıtlamıştır
- Manuel implementasyon algoritma mantığını kavramayı sağlar ancak üretimde kullanılmamalıdır

Sonuç olarak, modern güvenli iletişim sistemleri simetrik ve asimetrik şifrelemeyi birlikte kullanarak hem performans hem güvenlik açısından optimal çözüm sunmaktadır.